UIUCDCS-R-75-730

COMBINATORIAL LOGIC DESIGN OF A DIABLO SERIAL PRINTER CONTROLLER

by

Robert Lee Moulic

May, 1975

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN · URBANA, ILLINOIS

UIUCDCS-R-75-730

COMBINATORIAL LOGIC DESIGN OF A DIABLO SERIAL PRINTER CONTROLLER

BY

ROBERT LEE MOULIC

May, 1975

Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

Submitted in partial fulfillment for the requirements for the degree of Master of Science in Electrical Engineering, May 1975.

ACKNOWLEDGEMENT

I wish to gratefully acknowledge the assistance and guidance of Professor T. A. Murrell in the preparation of this thesis. I would also like to thank Professor Michael Faiman, Harold Lopeman, and Robert Skinner for their assistance during the experimental phase, and Mark Goebel and Zigrida Arbatsky for their assistance in the production of the final copy.

TABLE OF CONTENTS

1. INTRODUCTION

The purpose of this paper is to describe the desirability, background, design, and implementation of a digital controller for the Diablo Hytype I serial printer. The printer operates at 30 characters per second, has a 132 character print line capability, and utilizes an ASCII coding structure. Most of the existing computer terminals at the University of Illinois make use of at most 10 to 15 characters per second printing speed for printed characters. Other types are available which operate at 30 characters per second but they are of the wire matrix type which require a specially treated paper which costs significantly more than standard paper. In addition the legibility of the matrix characters is usually not of the highest quality. Since the Diablo prints in a normal fashion, it would be very ideally suited to use as a computer terminal. The Diablo, however, uses a nonstandard interface technique which requires the use of some additional hardware in order to use it as a teleprocessing terminal. In addition the Diablo possesses some additional functional capabilities which can be implemented in the controller with little additional expenditure of hardware facilities.

2. GENERAL DESIGN CRITERIA

The Diablo has a parallel interface structure. The various line designations are listed in Figure 1. All lines are negative logic; i.e., a 0 (0 volts) indicates that, for example, the printer is ready, while a 1 (5 volts) indicates that the printer is not ready. The important lines are briefly described below.

INPUT LINES

SELECT PRINTER enables all input and output lines.

DATA LINES provide the data input to the printer. For a character print operation the low order seven bits are the ASCII character to be printed. For a carriage operation the low order ten bits are the number of 1/60th inch increments that the carriage is to be moved while the high order bit is the direction of movement (0 for right - 1 for left). For a paper motion operation the low order ten bits are the number of 1/48th inch increments that the paper is to be moved (0 for upward - 1 for downward).

CHARACTER STROBE is a signal which initiates a print operation.

CARRIAGE MOTION STROBE is a signal which initiates a carriage operation.

PAPER FEED STROBE is a signal which initiates a paper movement operation.

RESTORE PRINTER causes the printer to reset internally and to move the carriage to the leftmost position.

INPUT/OUTPUT LINES

PIN	DESIGNATION
h	data 1
j	data 2
m	data 4
f	data 8
k	data 16
i	data 32
g	data 64
d	data 128
b	data 256
V	data 512
F	data 1024
P	character print strobe
K	carriage strobe
X	paper feed strobe
S	select printer
a	printer ready
Y	print ready
W	carriage ready
c	paper feed ready
A	ground
D	ground to be used as clock line

Figure 1. DIABLO INTERFACE LINE DESIGNATIONS

OUTPUT LINES

PRINTER READY indicates that the printer has power and is ready to operate.

PRINT WHEEL READY indicates that the printer is ready to accept a new character print operation.

CARRIAGE READY indicates that the printer is ready to accept a new carriage motion operation.

PAPER FEED READY indicates that the printer is ready to accept a new paper movement operation.

CHECK indicates a malfunction of the printer. This signal disables all other output lines and can be reset only by issuing a RESTORE PRINTER command.

There are certain minimum timings required for the interface signals in order for the printer to operate correctly. Figure 2 illustrates the timing dependencies. Different strobes may be applied a minimum of 400 nanoseconds apart. This effectively allows different commands to be overlapped. The timing between like strobes depends upon the execution time of the current command by the Diablo. Strobes must be a minimum of 1 microsecond in duration. Data lines must be maintained at least 200 nanoseconds prior to and after the strobe pulse leading and trailing edges, respectively.

The Diablo logic is composed almost exclusively of 5 volt TTL SSI and MSI devices. Since TTL devices are readily available and relatively inexpensive, it was decided that TTL logic would be used to implement the controller. Furthermore, where possible, the use of the same TTL packages in the controller as in the Diablo results in significant savings in spare parts stocking for repair purposes.

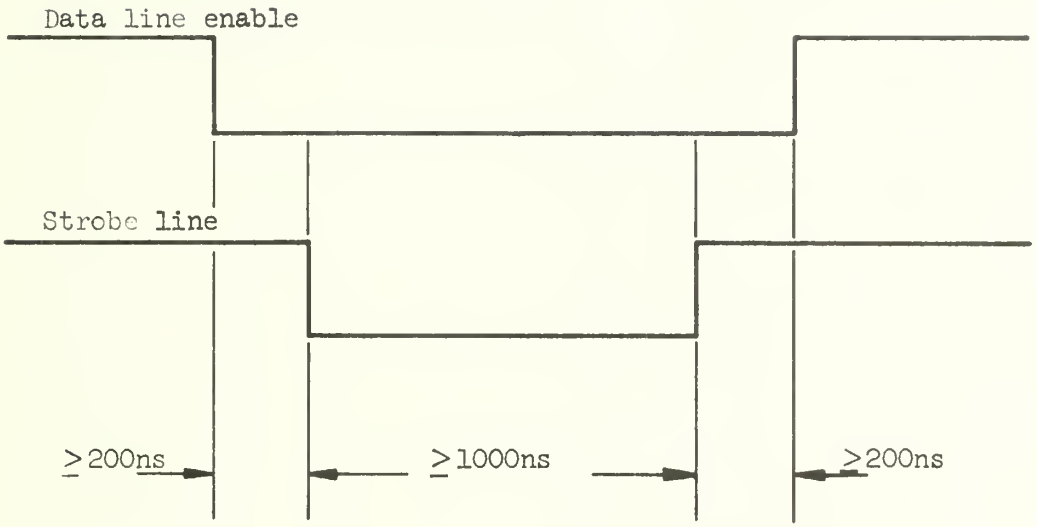
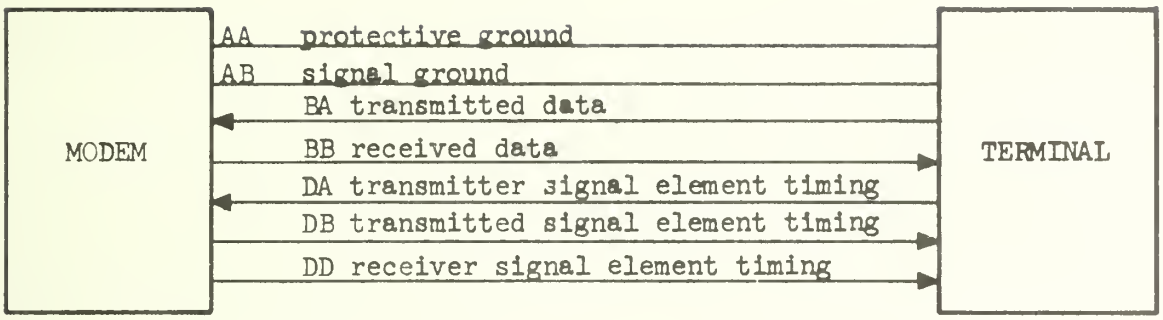
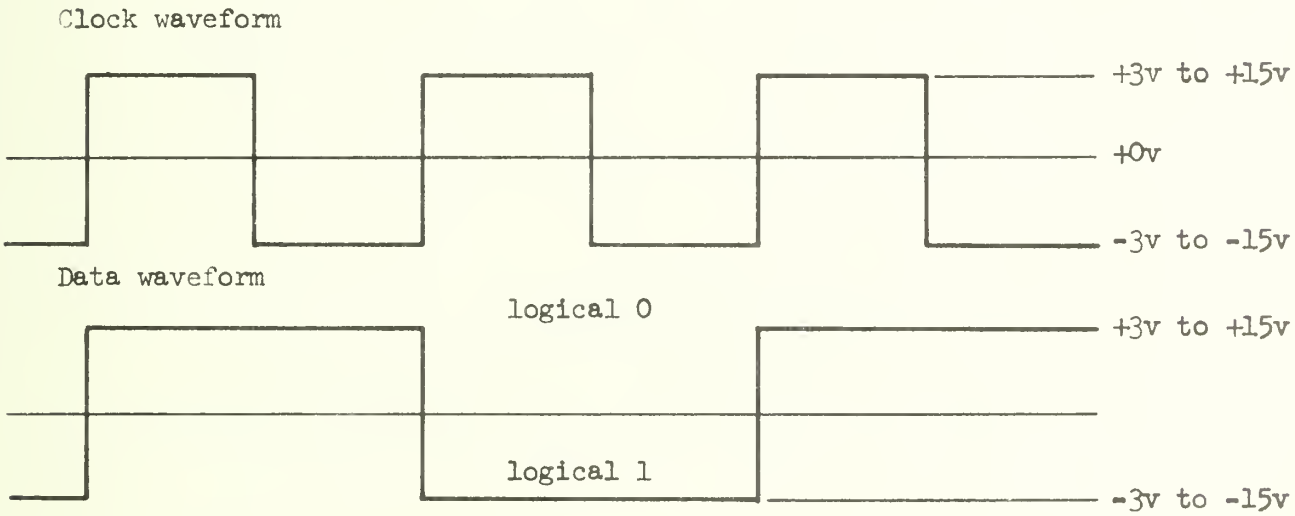


Figure 2. DIABLO INTERFACE SIGNAL TIMING REQUIREMENTS

Prototype implementation was carried out utilizing the Excel logic lab equipment from the CS 265 Logic Lab course. This resulted in some component substitutions when compared to the original ideal components design. Where substitutions were made, the alternative IC packages will be noted. The Excel approach proved to be a very practical solution to prototype design and construction, allowing rapid assembly and easy circuit changes to correct design errors. It is recommended that this approach be considered for other such projects in the future. As an aid to the advanced designer, some additional, high density cards should be designed to eliminate the "mass of spaghetti" maze of wires encountered in larger projects.

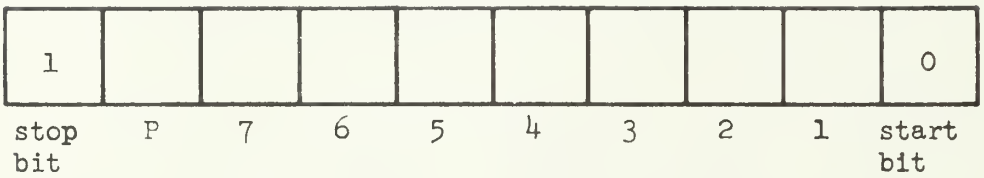
The standard EIA RS232C communications criteria is adhered to in the design of the controller. The RS232C interface is illustrated in Figure 3. Data transmission is at 300 Baud. The ASCII characters are transmitted bit serially in a synchronous manner, but characters are transmitted asynchronously with respect to other characters. Figure 4 illustrates the typical transmitted character.





NOTES:
1. Either DA or DB is used depending upon configuration.
2. Clock goes negative at center of data pulse.
3. Next data bit starts at positive going edge of clock.
4. Data waveform is negative logic.

Figure 3. EIA RS232C COMMUNICATIONS INTERFACE



NOTES:
1. Start is first bit transmitted.
2. Stop bit is last bit transmitted.
3. ASCII character is transmitted low order bit first.
4. P bit is parity bit.
5. Data line is held at logical 1 state between characters.

Figure 4. FORMAT OF TRANSMITTED ASCII CHARACTER AT 300 BAUD

3. CONTROLLER LOGICAL DESIGN

Operating within the constraints developed in Chapter 2, it was desirable to implement several enhancements to the normal TTY-like terminal device. It was decided to allow the Diablo to operate in a dual mode; text mode for normal TTY-like operation with enhancements, and incremental mode which allows complete utilization of all of the Diablo's capabilities.

In text mode all of the normal terminal functions are available such as print character, carriage return/line feed, and paper feed. Additional functions were added to perform subscript/superscript operations, wherein the platen is moved only half a line, thus permitting subscripting/superscripting such as appears in much of the technical literature. A reverse direction line feed was also implemented as a byproduct of this capability. An electronic tab feature was implemented to allow the print mechanism to move directly to any position from any position in one simple operation. One command is reserved to put the controller into the incremental mode.

In incremental mode, either one or two transmitted characters are required to perform a function. If the transmitted character is not the second character of a two character function, and if at least one of the two high order bits is a binary 1, corresponding to the ASCII printable character set, this indicates that the controller should perform a normal print function. If such is not the case, for the first character of a two character pair, the low order five bits of the character are stored in the controller. When the second character arrives at the controller,

it is effectively concatenated to the previously saved five bits and the appropriate function is performed. Additionally a character is reserved to return the controller to text mode.

With basic functions briefly outlined above, it is now possible to construct the controller in a high level, block diagram manner to illustrate data and control flow between the modules of the controller without having to be at present concerned with circuitry. Figure 5 illustrates the functional module interrelationships which are briefly described in the following paragraphs.

MODEM This is not a part of the controller but rather is the link between the controller and the telecommunications network. A Bell 103A or equivalent device is assumed.

KEYBOARD The keyboard is any TTL-compatible ASCII code generating with parallel data lines and strobe. The keyboard currently being used is a Cherry device.

INPUT INTERFACE The first function of the input interface is to convert the parallel input of the keyboard to the bit serial format illustrated in Figure 4. Then, if the Diablo is in local mode, this serialized character is transmitted to the deserializer portion of the interface. If the termial is in remote mode, the character is routed to the modem. The input deserializer converts the serial bit string to a parallel form in the input buffer. When a complete character has been assembled the input interface signals the input decoder that a new character has been assembled. Additonallly the 300 Hz clock is produced in this module.

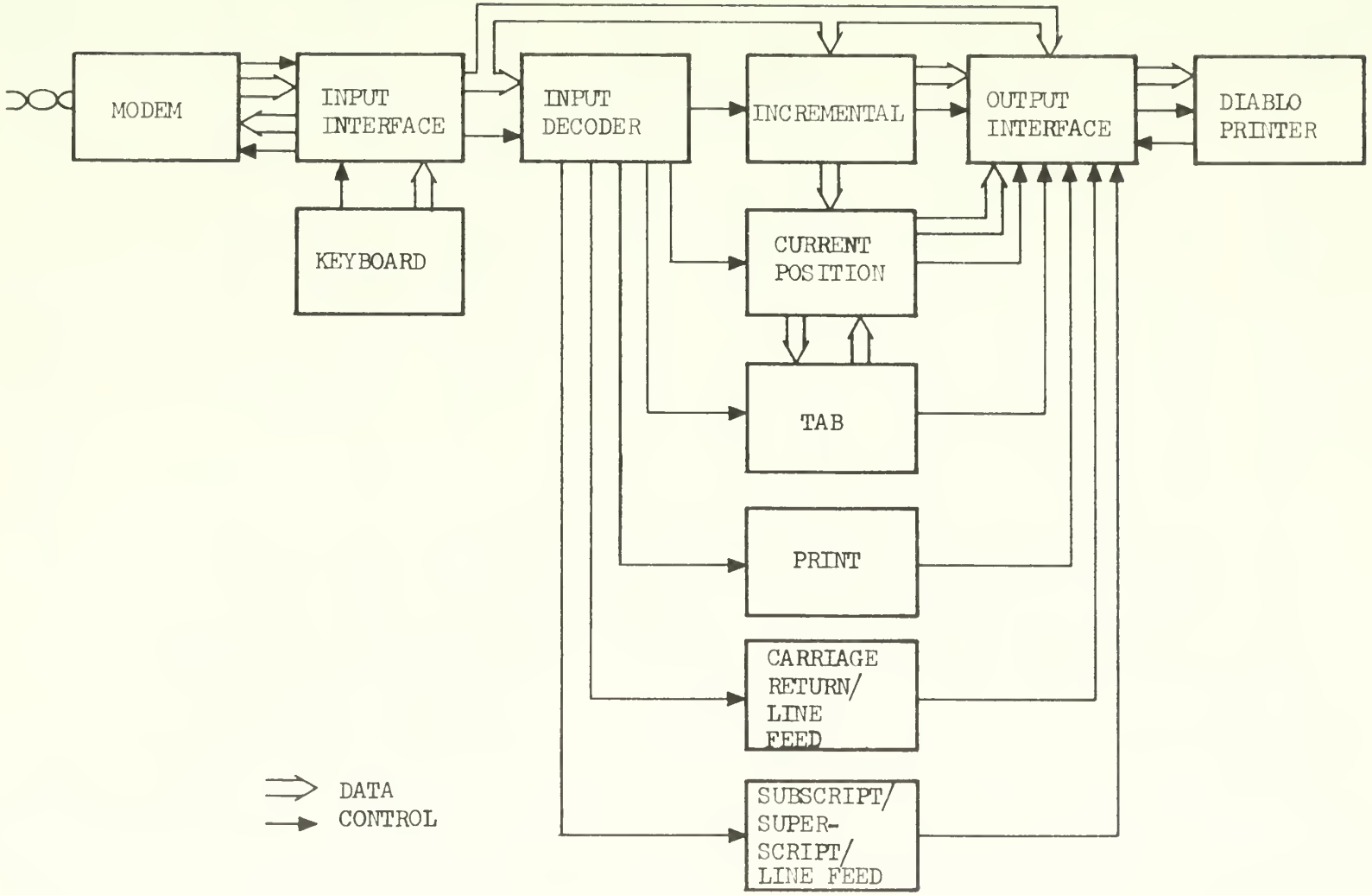


Figure 5. CONTROLLER BLOCK DIAGRAM

INPUT DECODER This module decodes the current input character and determines, utilizing the text or incremental mode status of the controller, what function, if any, should be performed. The decoder then signals the appropriate module to begin its cycle.

OUTPUT INTERFACE This module receives data and commands from various other modules and, based upon the information provided, causes the Diablo printer to initiate one of its functions.

INCREMENTAL This module contains all of the logic required to operate the Diablo when the controller is in incremental mode.

CURRENT POSITION This module contains a register whose contents represent the current position of the Diablo carriage. This value is used for carriage return operations and carriage limit-of-travel checking purposes.

TAB This module provides electronic tabulation and left or right tabbing motion during tabulation operations.

PRINT The print module contains the necessary logic to cause the Diablo to print a character and move the carriage one print position.

CARRIAGE RETURN/LINE FEED This module will control a carriage return or carriage return and line feed operation.

SUBSCRIPT/SUPERSCRIPT/LINE FEED This module implements the subscript, superscript, and line feed commands.

The following chapter will describe each of controller modules in detail. The complete controller schematic appears as Appendix A.

4. CONTROLLER CIRCUITRY FUNCTIONAL DESCRIPTION

4.1 INPUT INTERFACE MODULE

The input interface module performs all input functions for the controller. The 300 Hz clock is derived from the 480 kHz clock in the Diablo printer. One of the extra ground lines in the Diablo signal cable is relocated to the 480 kHz clock in the Diablo. The other end of the line in the controller is terminated at a 7414 Schmitt trigger which provides a noise immunity of about .8 volts. The 74290's and the 74293 step the 480 kHz signal down to 300 Hz.

The prototype controller uses 4 74194 4 bit shift registers, which were readily available in the Excel packages. A 74164 and 74165 combination would provide the same function in two fewer packages.

When the keyboard strobe goes high the output of NAND 8.1 goes high, thus loading the shift register. Simultaneously flip flops 8.1 and 8.2 are reset, 8.3 is set, and the 74193 counter is reset to zeroes. Flip flop 8.2 represents the 0 start bit while flip flop 8.3 holds the data line at a logical 1 between characters.

On the next clock cycle after the keyboard strobe goes low, the shift register begins shifting the data out onto the RS232C data line. Simultaneously the counter is incremented for each bit shifted. On the count of ten, which corresponds to the stop bit, the output of NAND 8.3 goes low and disables the shifting operation.

The LM1488's provide TTL to RS232 level conversion. The LOCAL/ REMOTE switch routes the data and clock either to the communications interface, which in the case of half duplex mode routes the data and clock back to the controller input lines, or directly to the LM1489 level converters in the case of local mode operation.

When a 0 bit appears on the data line in the start bit position, the output of NAND 9.1 goes low, resetting flip flop 9.1, which will allow the 74193 counter to begin to count. The data is serially shifted into the shift register on Figure 9 while the counter is simultaneously, but one count behind, incrementing. On the count of 7 the output of NAND 9.3 goes low thus providing the reset pulse to the input decoder module. On the count of 8 the 8 data bits have been assembled in the shift register (the start bit has propagated out of the shift register) and the input buffer is clocked, transferring the contents of the shift register into the input buffer.

On the count of 9 the output of NAND 9.2 goes low resetting flip flop 9.1 which resets and inhibits the counter and provides the clock edge to trigger the input decoder module operation.

Parity checking has been left out intentionally. The incidence of errors at 300 baud is extremely low and the University of Illinois Plorts system does not implement parity checking at the Transmission control unit end of the telecommunications link.

4.2 INPUT DECODER MODULE

The input decoder module is essentially a multiple output network composed of NANDs and inverters and a command register of flip flops

which retain the outputs of the network at a specific instant in time. In addition the decoder contains the controller mode flip flop whose state determines whether the controller is in text or incremental mode.

At a count of 7 in the 74193 of Figure 9 the command register is reset to zeroes. Some of the flip flops may actually be set to one depending upon which of the outputs, Q or $\overline{Q}$, is utilized. At a count of 8 the new input buffer contents are applied to the decoding network. At a count of 9, well after the network outputs have stabilized, the clock signal is applied to the command register.

The change in state of one of the command flip flops is then used to provide the positive edge clock signal used to initiate a specific function. If the input is a null character or SET MODE command, none of the command flip flops is set.

Figure 6 illustrates the binary data patterns and their corresponding functions. Boolean algebra has been applied to the decoder network to facilitate its construction from readily available NAND gates.

4.3 INCREMENTAL MODULE

The incremental module controls the Diablo printer in its fundamental mode of operation. Nonprinting operations require the effective concatenation of two transmitted data characters to perform one Diablo command. Flip flops 12.1 and 12.2 and NAND 12.1 determine what command is to be performed in the incremental mode.

When a SET INCREMENTAL MODE command occurs, flip flop 12.1 is set. When the next character is received by the controller, flip flops 12.1 and 12.2 will be clocked. Flip flop 12.2 will reset. If a printable

BIT PATTERN	MODE	CONTROLLER FUNCTION
0000000	text	null character
0000001	text	set incremental mode
0000100	text	clear tab
0000101	text	set tab
0000110	text	tab right
0000111	text	tab left
0001000	text	back space one line
0001001	text	superscript
0001010	text	line feed
0001011	text	subscript
0001100	text	carriage return
0001101	text	carriage return/line feed
0001111	incremental	set text mode
000XXXX	incremental	first character of carriage command
001XXXX	incremental	first character of paper command

Figure 6. CHARACTER BIT PATTERNS AND CORRESPONDING FUNCTIONS

character is present in the input buffer (bit 1 or 2 is a logical one), flip flop 12.1 will remain set and printing activity will be controlled by the print module. If a nonprintable character is present, indicating an incremental function, flip flop 12.1 will reset, causing the low order 5 bits of the character in the input buffer to be stored in flip flops 12.3 thru 12.7, respectively. On the arrival of the second character, flip flop 12.1 will set and flip flop 12.2 will set also.

The setting of flip flop 12.2 simultaneously gates flip flops 12.4 thru 12.8 onto the high order output data register input bus lines, gates the new data in the input buffer onto the low order output data register input bus lines, and signals the output interface to perform either a carriage or paper command, depending upon the setting of flip flop 12.3. The output interface then resets flip flop 12.2, thus preparing the incremental module for the next operation.

4.4 PRINT MODULE

The print module controls all print functions of the controller. The module functions in a space after print mode of operation. When a print command is decoded, flip flop 13.1 is set, signalling the output interface to perform a print function while, at the same time, gating the character to be printed onto the low order output data register input bus lines. The high order output data register bits are ignored by the Diablo on a print function but are nonetheless set to zeros by the controller. After accepting the print command the output interface resets flip flop 13.1.

At the termination of the print phase the output interface clocks flip flop 13.2. Since the J lead was a one the flip flop will set,

initiating a carriage movement function. The carriage will be moved right a binary count of 110, which is one tenth inch, corresponding to ten characters per inch. At the termination of the carriage phase, the output interface again clocks flip flop 13.2. Since the K lead now is one flip flop 13.2 will reset, returning the print module to its original state.

The input buffer to output data register gating logic is also utilized by the incremental module.

4.5 CARRIAGE RETURN/LINE FEED MODULE

The carriage return/line feed module performs the standard ASCII CR and CR/LF functions. When a CR function is decoded flip flop 14.1 is set, resetting the carriage motion flip flop (current position module), gating the current position register contents onto the low order output data register input bus lines, gating a logical one into the high order position of the output data register to indicate leftward movement, and signalling the output interface to perform a carriage function. After the carriage command is accepted by the output interface, flip flop 14.1 will reset at the end of the carriage motion cycle, thus removing the reset pulse from the current position register.

If this is a CR/LF command, flip flop 14.2 will be clocked. Since the J lead was a one flip flop 14.2 will set, signalling a paper function to the output interface, and applying a binary count of 1000 to the output data register, which results in a one sixth inch movement of the paper. At the completion of the paper cycle flip flop 14.2 will be reset.

The implementation of carriage return prior to line feed allows an effective overlap of the two functions thus reducing total command time. Maximum carriage return time is 400 milliseconds for 13.2 inches. Sufficient nulls or idles should be inserted in character streams to the terminal to allow for carriage return time.

4.6 SUBSCRIPT/SUPERSCRIPT/LINE FEED MODULE

This module provides an alternative to integral, one direction paper motion. When a script/line feed command is decoded, flip flop 15.1 sets, signalling the output interface that a paper command is to be performed. If the command is a script command a binary count of 100 is applied to the output data register, indicating a one twelfth inch paper movement. If the command is a line feed a one sixth inch increment is applied, indicating a normal line spacing width. The direction is applied to output data register position 1024, allowing paper motion in either direction.

When the output interface accepts the command, flip flop 15.1 is reset.

4.7 OUTPUT INTERFACE MODULE

The output interface module directly controls the Diablo printer, supplying the data, appropriate strobes, and correct timing sequences. When one of the function modules signals the output interface to perform a function, flip flop 16.1 is set, generating the SELECT PRINTER signal to the Diablo and clocking the output data register (flip flops which

appear on Figure 17). The Diablo then replies by lowering the ready lines for the functions it is capable of performing at the moment.

The combination of the ready signal and the function signal sets flip flop 16.2 and one of the strobe enable flip flops, 16.3, 16.4, or 16.5, depending upon which function is to occur. The setting of flip flop 16.2 triggers one shot 16.1, which has a pulse width of at least 200 nanoseconds, corresponding to the data stabilization of Figure 2. Flip flop 16.2 also is used to enable the data lines to the Diablo. The trailing edge of the one shot 16.1's pulse is used to trigger one shot 16.2, which provides the 1 microsecond strobe, and also resets flip flop 14.3 during a carriage return, thereby resetting the current position register. The trailing edge of one shot 16.2 triggers one shot 16.3, which provides the 200 nanosecond data hold time and resets the various function initiation flip flops. The trailing edge of one shot 16.3 triggers one shot 16.4, which resets the output interface logic. Note that a new function is prevented from beginning until flip flop 16.1 is reset by one shot 16.4.

The output register input NANDs are effectively OR gates realized via an application of DeMorgan's theorem. The output NANDS provide logic inversion for the negative logic levels required by the Diablo printer.

4.8 CURRENT POSITION MODULE

The current position module contains all logic required to determine if a carriage operation will not exceed the mechanical limits of travel of the Diablo mechanism. The contents of the current position register (Figure 19) represent the current position of the carriage

mechanism. The value will also be used for carriage return operations and the tab functions.

Whenever a carriage motion function is initiated the incremental value to be sent to the Diablo is applied as the augend input to a modified sign magnitude adder. If the motion is leftward the 1024 bit position is a one. This one, when applied as the second input to the augend Exclusive OR's, complements the augend value. The addend input to the adder is the contents of the current position register. The adder output is then sampled to determine if the anticipated carriage movement would exceed the limits of motion. If such is the case a logical one appears at the D input of the inhibit flip flop.

After the printer has been successfully selected for the carriage operation, the output interface clocks the carriage inhibit flip flop (Figure 18). It is important that the actual adder packages chosen be fairly fast to ensure that the flip flop D value has stabilized before the clock input arrives. Depending upon available devices, it may be necessary to delay the clock pulse via a one shot or perhaps several daisy chained inverters.

If the D input is a zero, indicating a valid motion, the inhibit flip flop is reset, thereby enabling the carriage position strobe and clocking the current position register to store the new current position value, which is the output of the adder. In this case it is important that the data hold time of the flip flops is less than propagation delay time of the flip flops and adders. Normally this is not a problem. If the D input is a one, the flip flop is set, the carriage strobe is inhibited, and the new sum is not gated into the current position register.

During a carriage return the inhibit flip flop is asynchronously reset and the current position register is reset to zeroes after it is gated into the output data register.

Note that recomplementing circuitry for the adder sum is not necessary since only positive sums are of interest. Also, the adder is free running with a meaningful output only immediately prior to the inhibit flip flop clock edge. The print width has been limited to 128 character positions for reasons of simplicity. The full 132 positions can be provided.

4.9 TAB MODULE

The tab module provides electronic tab functions for the Diablo printer. One command moves the carriage mechanism directly to a tab set position in either a left or right direction. There is no "pseudo" tab wherein a series of space commands is performed. A 1024 bit random access memory is utilized for tabbing purposes. When a bit contains a one no tab is set at that carriage position; a zero bit indicates a tab has been set at that position.

When a tab command is decoded, the RAM address register is loaded with the contents of the current position register. If the command is also a clear command the address register is cleared, thus overriding the load command. As this is occuring the tab register (Figure 21) is cleared.

At the end of the 250 nanosecond one shot 20.1 pulse one shot 20.2, which has a one microsecond pulse corresponding to the cycle time of the RAM, is triggered initiating the RAM cycle. If the operation is a clear or set tab the RAM is put in a write mode. If this is a tab left

or right command the RAM is in read mode. The appropriate input data is applied for a write operation.

At the trailing edge of the one shot 20.2 pulse one shot 20.3, which has a 250 nanosecond pulse, may be triggered. If the RAM output is zero or if the address register has exceeded the allowable maximum or has become negative, one shot 20.3 will not trigger and the tab cycle is complete. If such is not the case the address register will increment for clear and tab right or decrement for tab left and the tab register will increment. At the trailing edge of the one shot 20.3 pulse one shot 20.2 will be triggered therby repeating the cycle until one of the stop conditions is encountered.

If a zero is read from the RAM and the command is a tab right or left command flip flop 20.3 will be set, initiating a carriage operation and gating the tab register contents onto the output data register input bus lines. Since this is a test before increment operation, if the current carriage position has a tab set a tab right or left command will result in no motion of the carriage.

Flip flops 20.1 and 20.2 store required function data in the event that long tab operations might cause overruns and require null or idle characters to be sent to the terminal.

5. CONTROLLER FABRICATION

The controller is intended to be fabricated in modular form with physical modules corresponding to the logical modules described in previous chapters. This would result in maximum serviceability at the expense of a large number of boards (nine or more) and inability to fully utilize all components in each integrated circuit package (thirteen extra packages are required in this method as compared to the minimum solution). Many boards would also be sparsely populated while others would be very densly packed. At the other extreme, one large board would save thirteen packages but would be very difficult if not impossible to service without wholesale disassembly of the board.

A practical solution is to combine some of the logical modules into a smaller number of physical modules. Only three extra I.C. packages are required with this approach. The module combinations and number of packages each are: input interface and input decoder, 31 packages; output interface, 24 packages; print, carriage return/line feed, and current position modules, 19 packages; and incremental, line feed/subscript/superscript, and tab modules, 23 packages. This approach yields manageable board size, nearly uniform board population, improved serviceability, and near minimum package count.

As an additional benefit of this approach, an "economy" model controller could be built by simply not plugging in the last module mentioned above. Since all of this module's outputs are active

when low lines, the controller will still function but without the TTY enhancements. This would effect an approximate cost savings of forty-five dollars for the I.C. packages alone.

Figure 7 lists package types, costs, and supply current requirements. Numbers in parentheses are maximum and minimum package counts for the first two construction methods. Prices are for lots of one and thus represent a high limit on prices.

QUANTITY	P/N	DESCRIPTION	SUPPLY CURRENT(ma) TYP	MAX	COST
20(22/20)	7400	quad 2 inp NAND	240	440	11.60
5(7/4)	7404	hex inverter	90	165	3.60
8(10/7)	7410	triple 2inp NAND	72	132	4.64
1	7414	hex Schmitt trigger	39	60	4.30
2	7420	dual 4inp NAND	12	22	1.16
1	7427	triple 3inp NOR	10	26	.84
7	7430	8inp NAND	21	42	4.06
10(12/10)	7474	dual D flip flop	170	300	8.90
2(3/2)	7476	dual JK flip flop	40	80	2.12
3	7486	quad EOR	90	150	2.79
7	74121	one shot	161	210	7.91
4	74174	quad D flip flop	180	260	17.12
6(7/5)	74175	hex D flip flop	180	270	21.84
8	74193	4 bit up/down counter	520	816	31.44
4	75194	4 bit shift register	156	252	12.20
3	74283	4 bit adder	198	330	8.28
2	74290	decade counter	58	84	3.20
1	74293	binary counter	26	39	3.20
1	LM1488	line driver(15v supply)	50	68	3.55
1	LM1489	line receiver	14	26	6.75
1	2102	1024 bit static RAM	30	60	15.00
TOTALS					
97(107/94)			2307	3764	$172.90

Figure 7. CONTROLLER INTEGRATED CIRCUIT REQUIREMENTS

LIST OF REFERENCES

Diablo Systems Inc., Model 1200 Hytype L Printer Maintenance Manual, n. publ., Hayward, California, 1972

Marcus, Mitchell P., Switching Circuits for Engineers, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1967.

Morris, Robert L., and Miller, John R., eds., Designing with TTL Integrated Circuits, McGraw-Hill, New York, 1970.

Muroga, S., "Class notes for Computer Science 391 - Logical Design and Seitching Theory," unpublished, University of Illinois at Urbana-Champaign, Urbana, Illinois, 1974.

National Semiconductor Inc., Digital Integrated Circuits, n. publ., Santa Clara, California, 1973.

Robertson, James E., "Class notes for Computer Science 394 - Introduction to Digital Computer Arithmetic," unpublished, University of Illinois at Urbana-Champaign, Urbana, Illinois, 1973.

Texas Instruments Inc., The TTL Data Book for Design Engineers, n. publ., Dallas, Texas, 1973.

APPENDIX

This appendix contains the entire set of controller logic diagrams. They are grouped together to provide a convenient reference for anyone who wishes to analyze the controller in great detail.

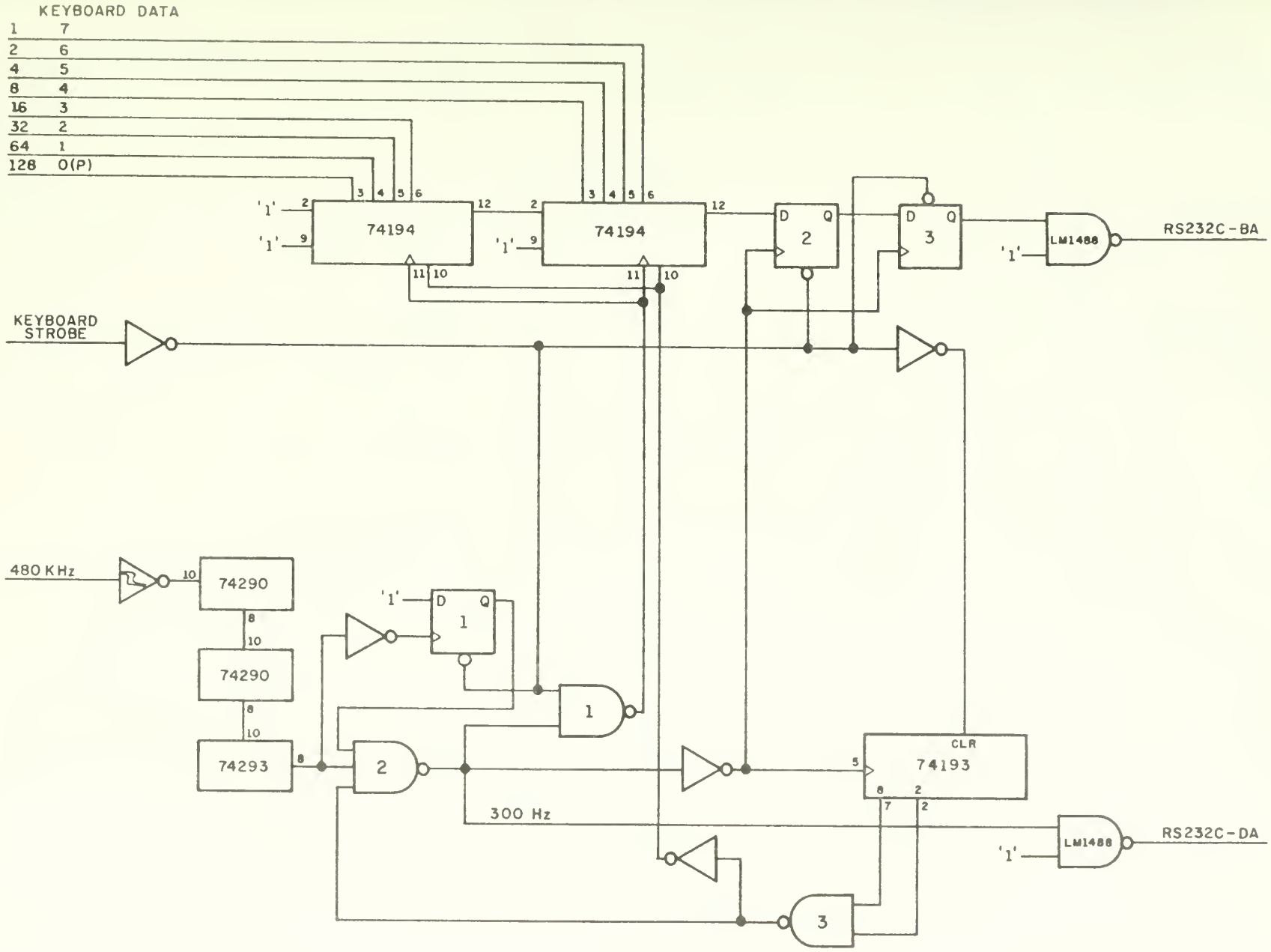


Figure 8. INPUT INTERFACE

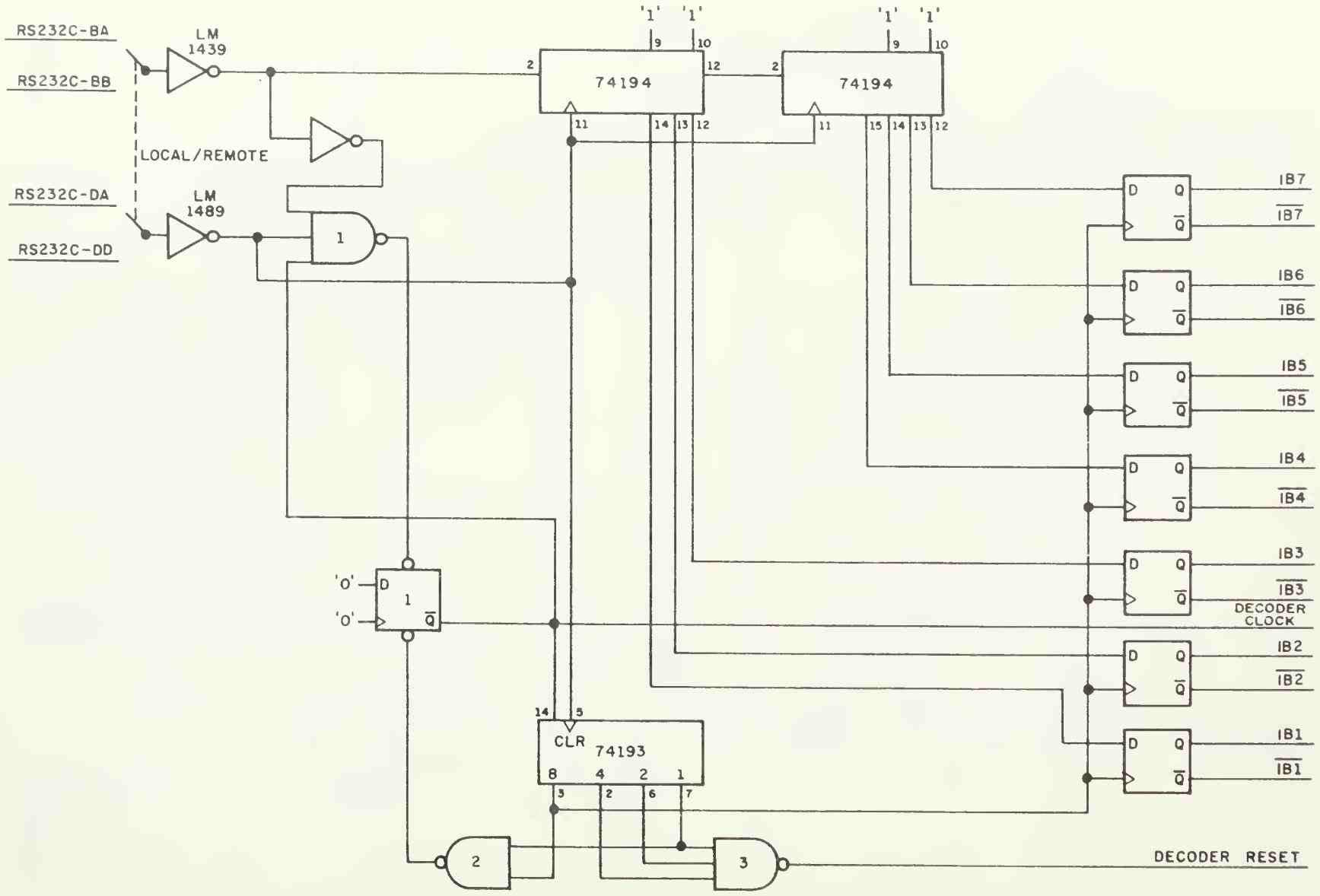


Figure 9. INPUT INTERFACE

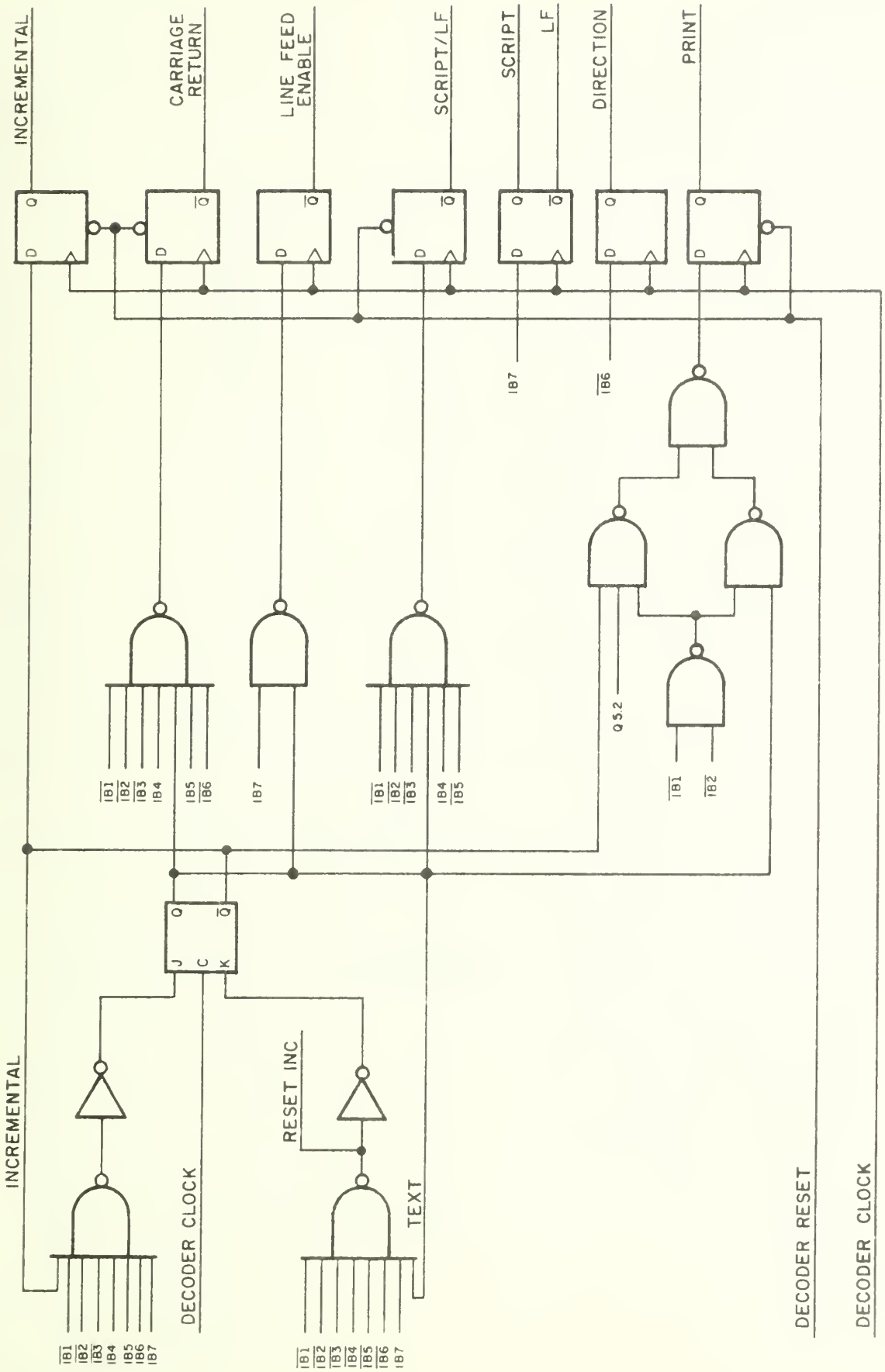


Figure 10. INPUT DECODER

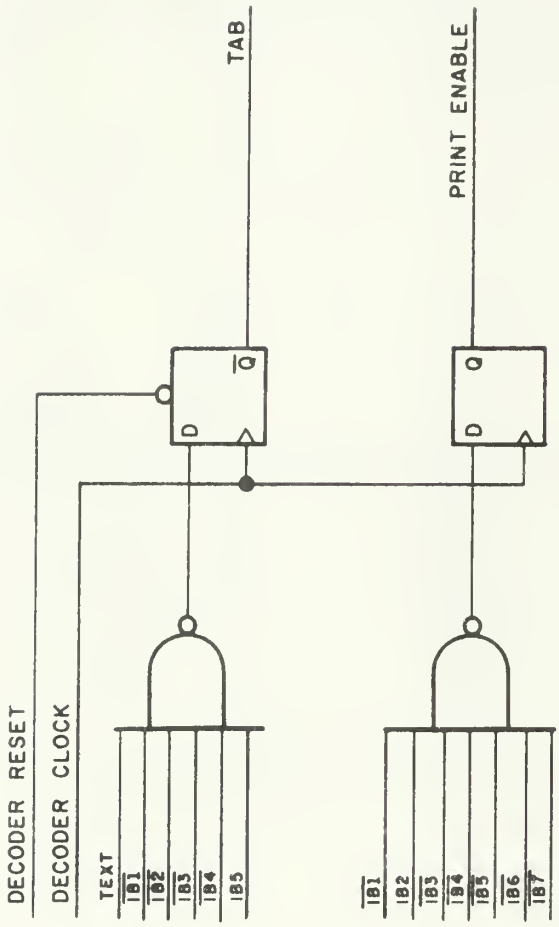


Figure 11. INPUT DECODER

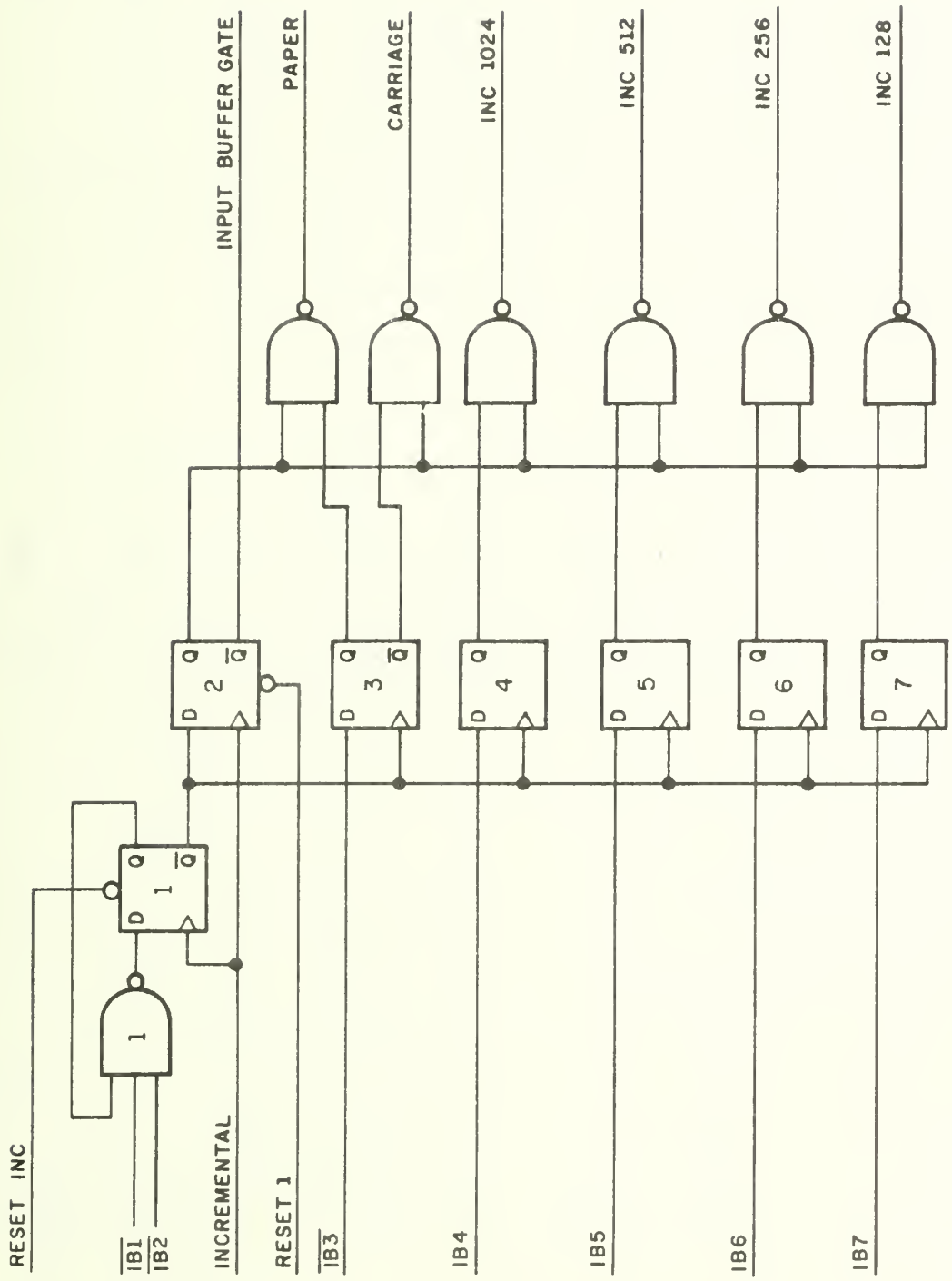


Figure 12. INCREMENTAL MODULE

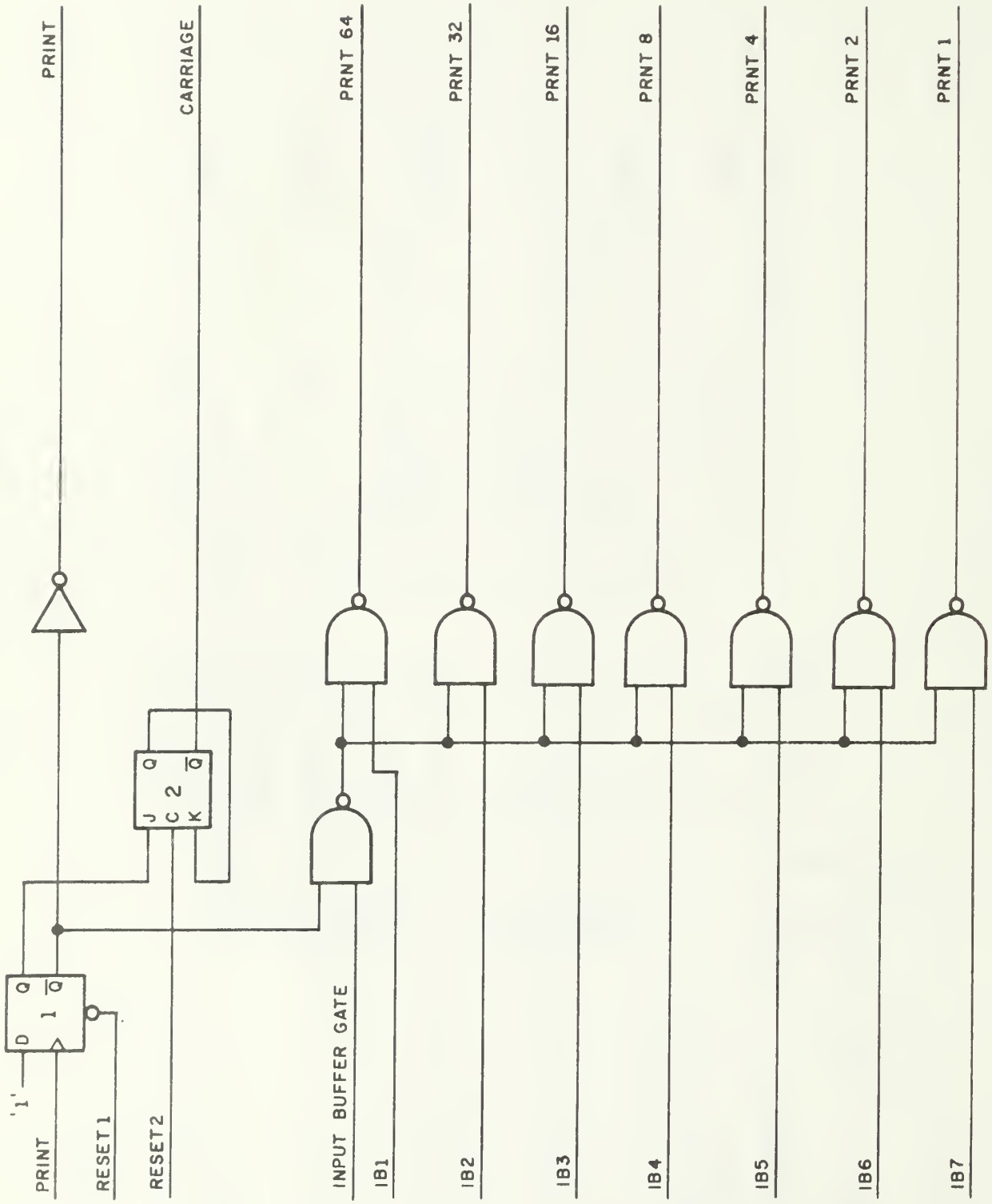


Figure 13. PRINT MODULE

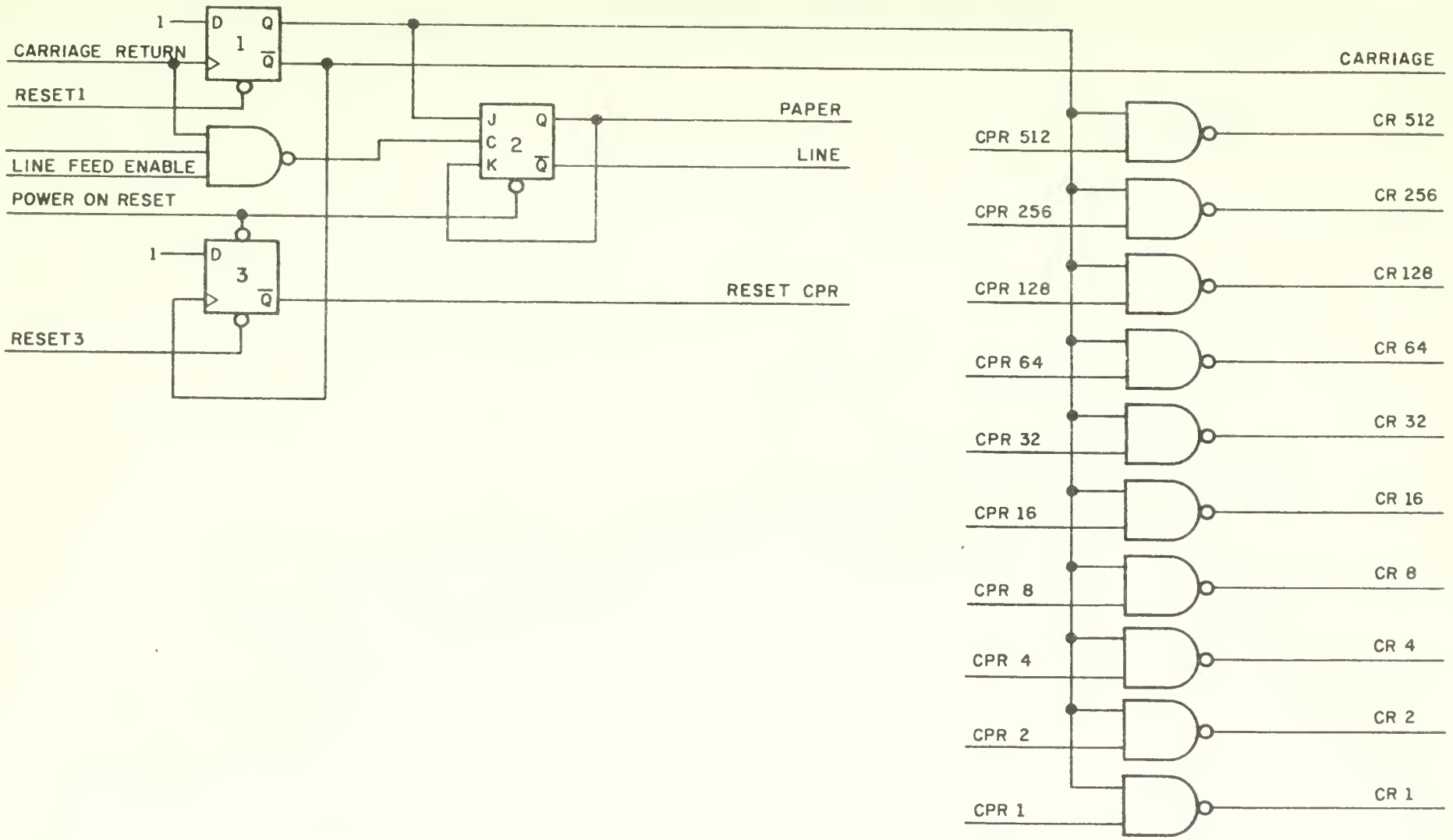


Figure 14. CARRIAGE RETURN/LINE FEED MODULE

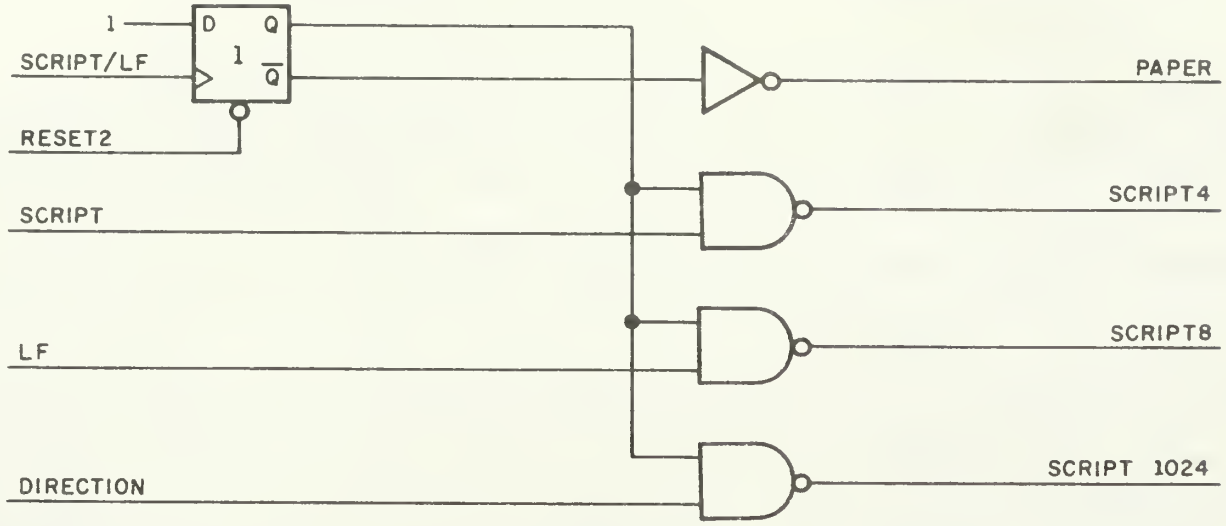


Figure 15. SUBSCRIPT/SUPERSCRIPT/LINE FEED MODULE

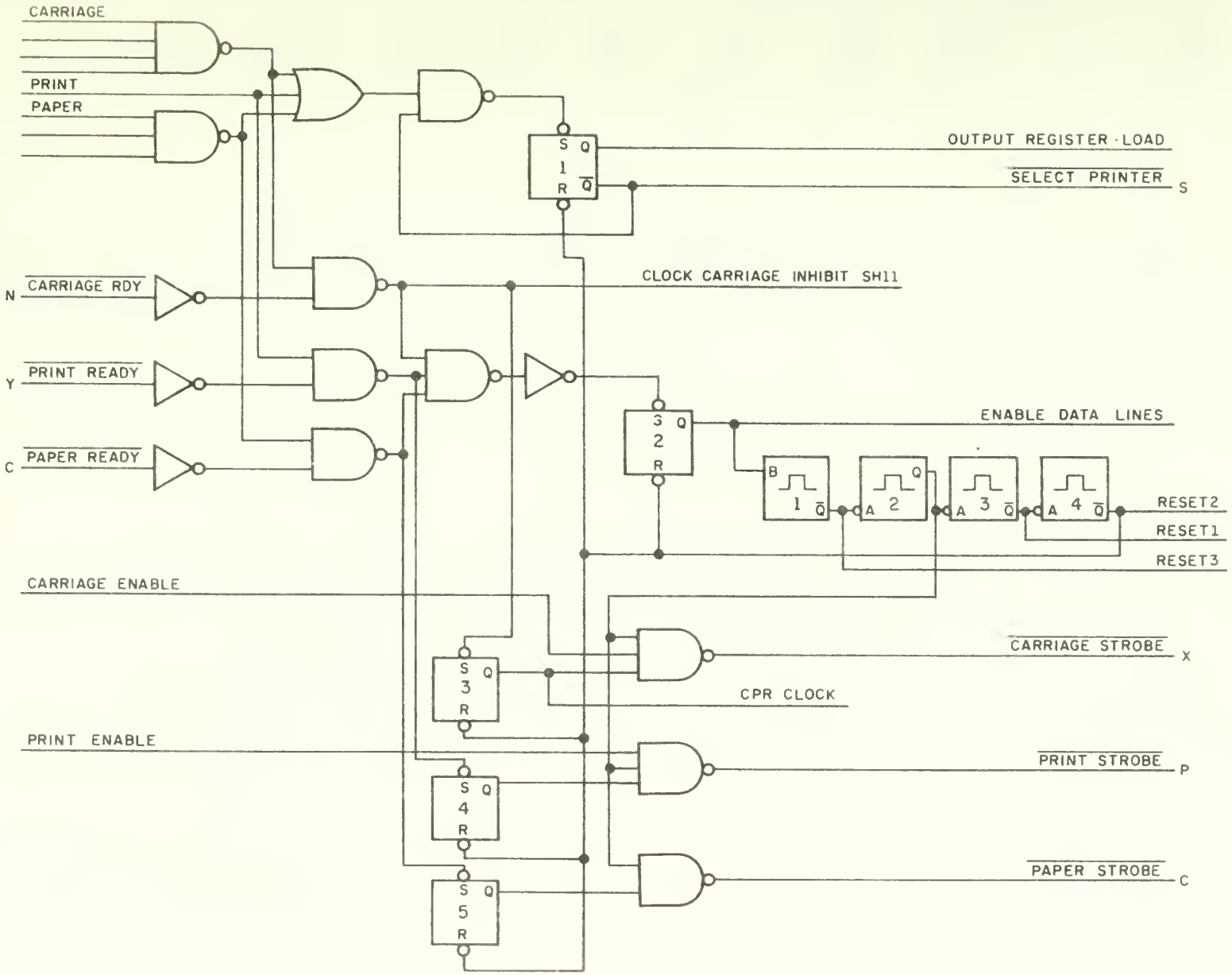


Figure 1c. OUTPUT INTERFACE MODULE

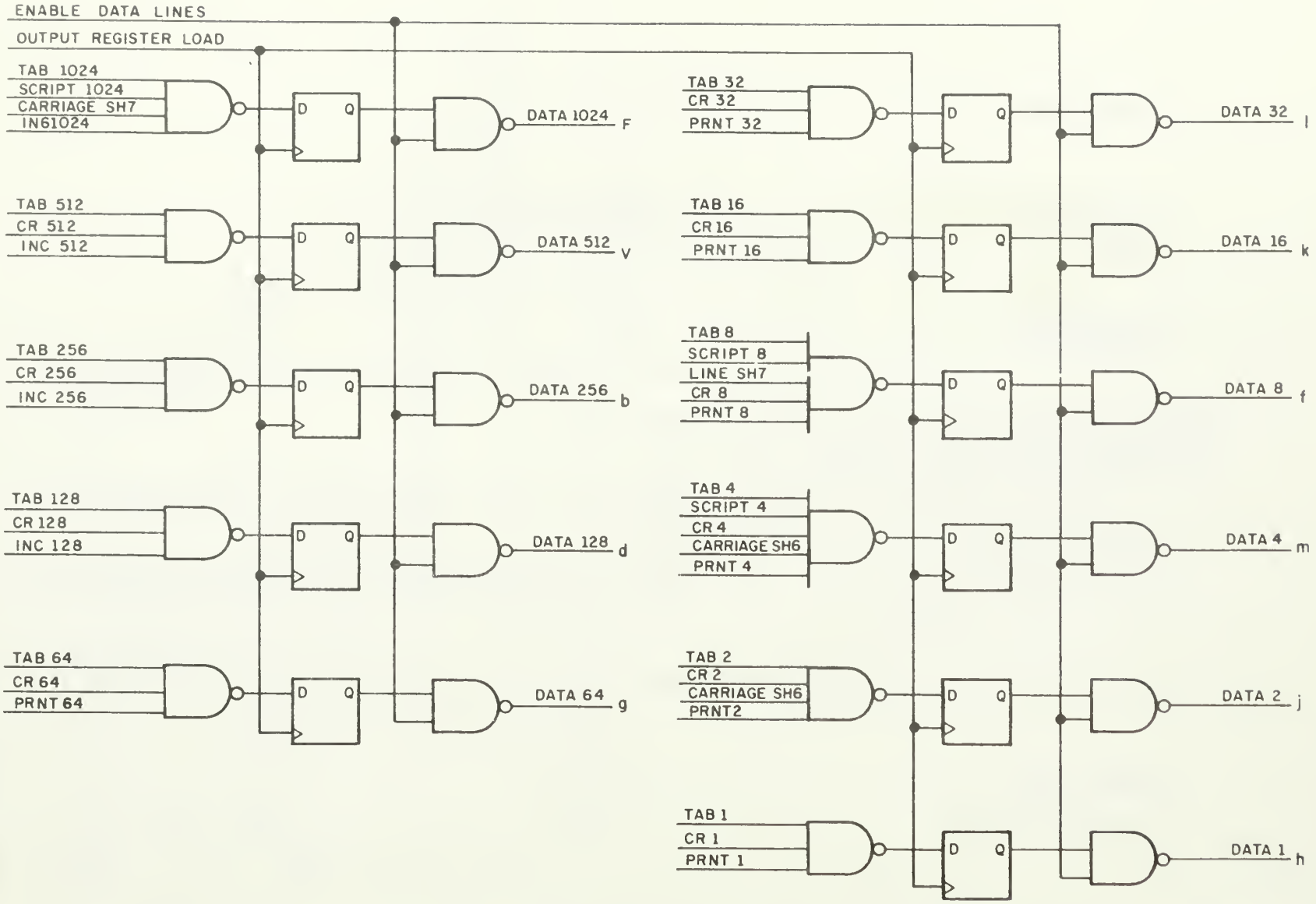


Figure 17. OUTPUT INTERFACE MODULE

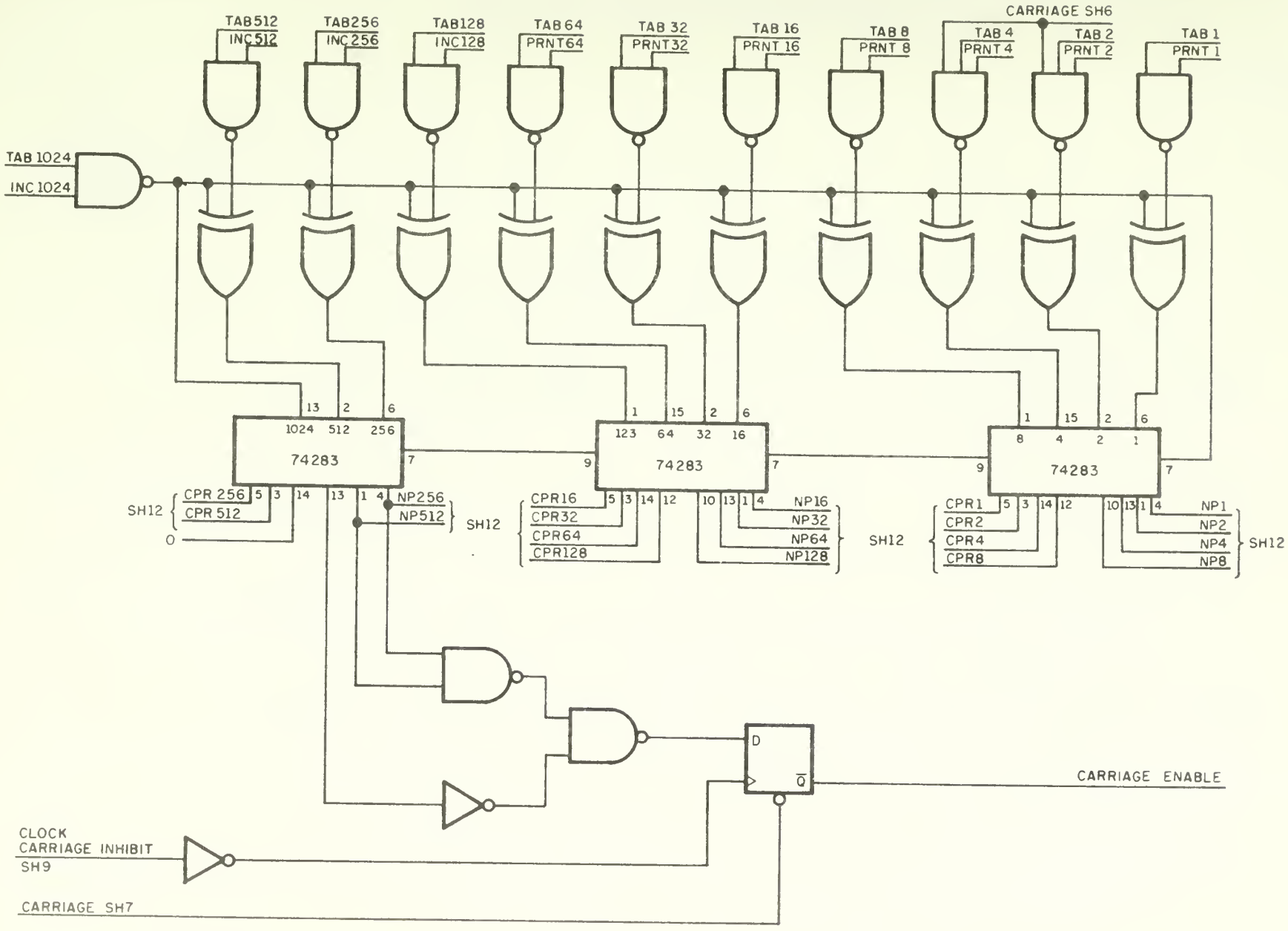


Figure 18. CURRENT POSITION MODULE

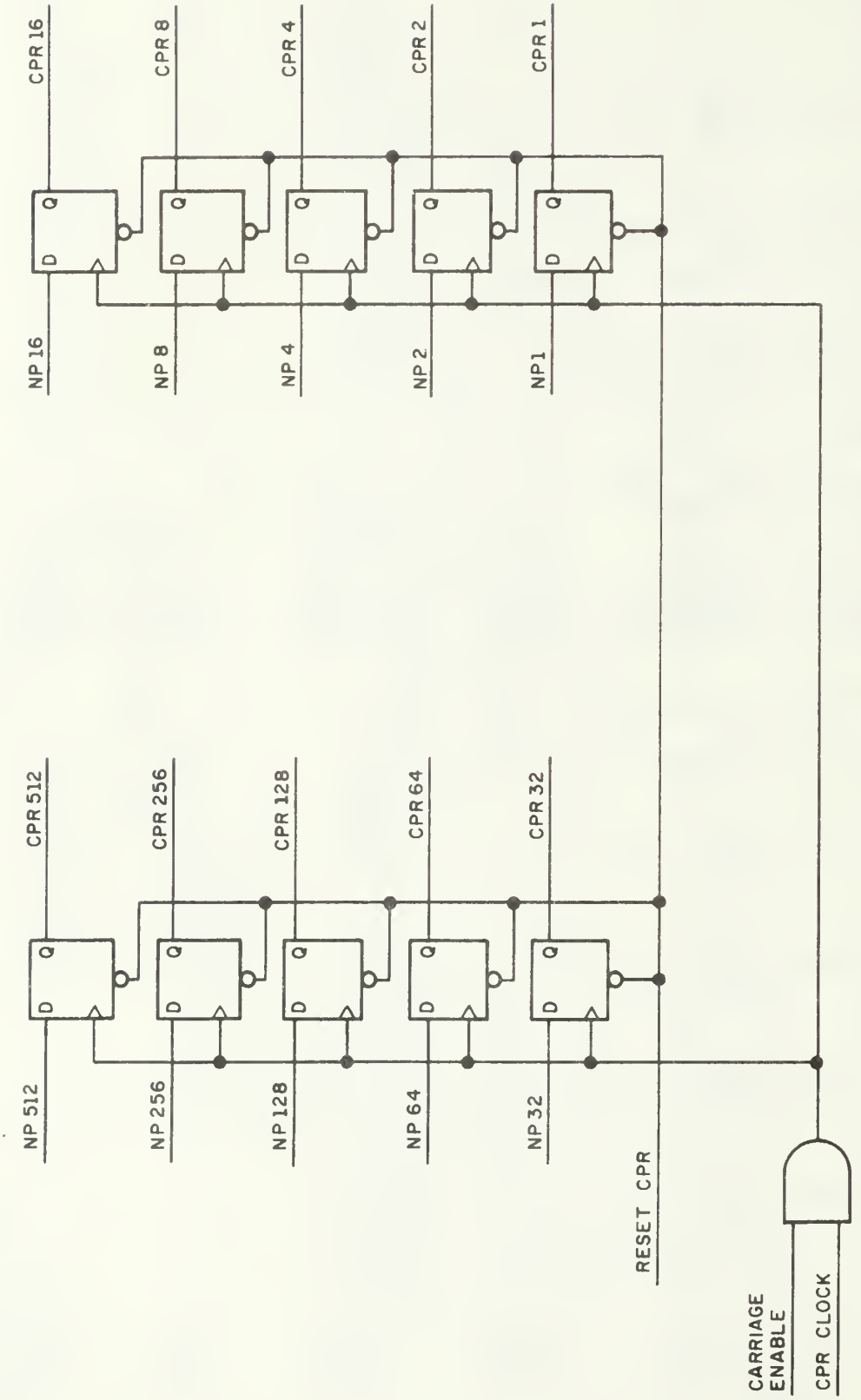


Figure 19. CURRENT POSITION REGISTER

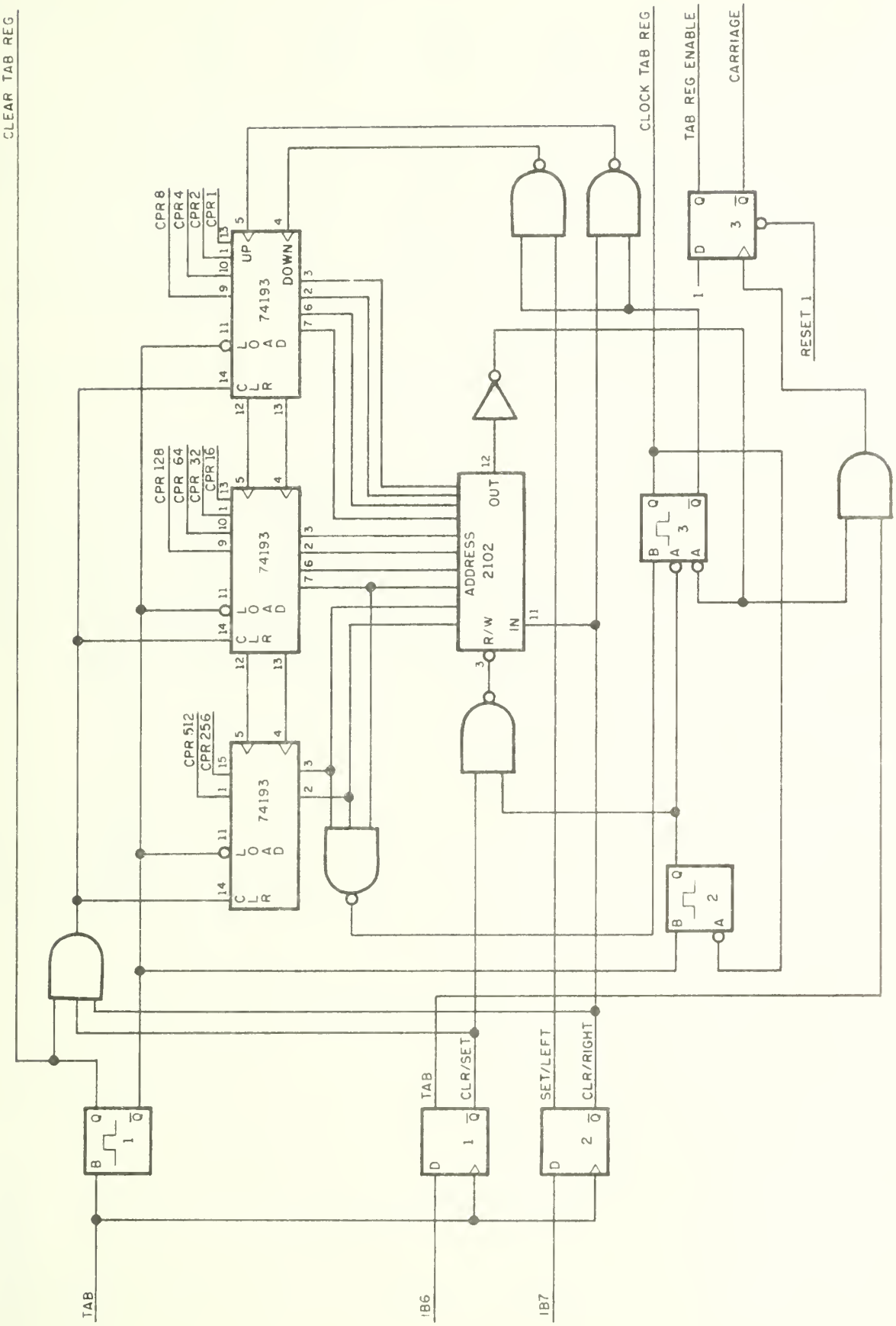


Figure 20. TAB FUNCTION MODULE

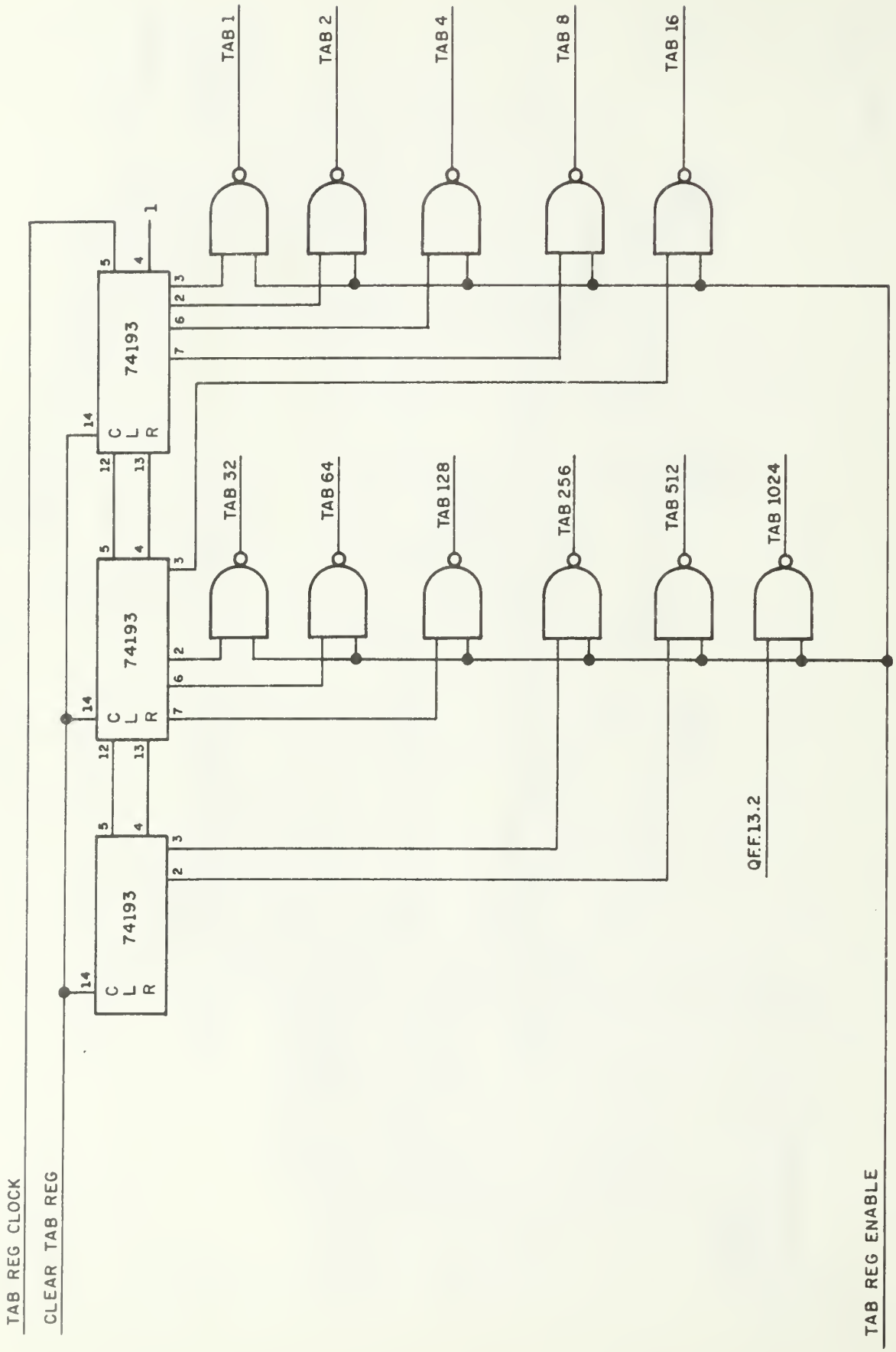


Figure 21. TAB FUNCTION MODULE

BIBLIOGRAPHIC DATA SHEET	1. Report No. UIUCDCS-R-75-730	2.	3. Recipient's Accession No.

4. Title and Subtitle
COMBINATORIAL LOGIC DESIGN OF A DIABLO SERIAL PRINTER CONTROLLER

5. Report Date
May, 1975

6.

7. Author(s)
Robert Lee Moulic

8. Performing Organization Rept. No. UIUCDCS-R-75-730

9. Performing Organization Name and Address
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

10. Project/Task/Work Unit No.

11. Contract/Grant No.

12. Sponsoring Organization Name and Address

13. Type of Report & Period Covered
Technical

14.

15. Supplementary Notes

16. Abstracts

This thesis describes the design and implementation of a controller for the Diablo Hytype I serial printer. The objective of the project is to develop a low cost thirty character per second ASCII computer terminal. In addition several enhancements, such as electronic tabulation and incremental mode of operation, are implemented. The controller is designed in modular form utilizing TTL components to facilitate design changes and improvements.

17. Key Words and Document Analysis. 17a. Descriptors

Computer Terminal
Terminal Controller
Start/stop terminal
RS-232 compatible terminal

17b. Identifiers/Open-Ended Terms

17c. COSATI Field/Group

18. Availability Statement	19. Security Class (This Report)	21. No. of Pages
RELEASE UNLIMITED	UNCLASSIFIED	46
	20. Security Class (This Page) UNCLASSIFIED	22. Price